Acta Astronautica 102 (2014) 367–377

Contents lists available at ScienceDirect

Acta Astronautica

journal homepage: www.elsevier.com/locate/actaastro

A near-optimal guidance for cooperative docking maneuvers

Marco Ciarcià *, Alessio Grompone, Marcello Romano

Mechanical and Aerospace Engineering Department, Naval Postgraduate School, 700 Dyer Road, Monterey, CA 93943, United States

ARTICLE INFO

Article history:
Received 26 June 2013
Received in revised form
2 December 2013
Accepted 6 January 2014
Available online 14 January 2014

Keywords:
Near-optimal guidance
Docking maneuver
Direct method
Real-time optimization
Rendezvous
Nonlinear programming

ABSTRACT

In this work we study the problem of minimum energy docking maneuvers between two Floating Spacecraft Simulators. The maneuvers are planar and conducted autonomously in a cooperative mode. The proposed guidance strategy is based on the direct method known as Inverse Dynamics in the Virtual Domain, and the nonlinear programming solver known as Sequential Gradient-Restoration Algorithm. The combination of these methods allows for the quick prototyping of near-optimal trajectories, and results in an implementable tool for real-time closed-loop maneuvering. The experimental results included in this paper were obtained by exploiting the recently upgraded Floating Spacecraft-Simulator Testbed of the Spacecraft Robotics Laboratory at the Naval Postgraduate School. A direct performances comparison, in terms of maneuver energy and propellant mass, between the proposed guidance strategy and a LQR controller, demonstrates the effectiveness of the method.

1. Introduction

In a previous paper [1], we have introduced a method to rapid prototype near-optimal planar proximity maneuvers. It was based on the coupling of a direct method, the Inverse Dynamics in the Virtual Domain (IDVD), and a nonlinear programming solver, the Sequential Gradient-Restoration Algorithm (SGRA). The maneuver scenarios required a chaser spacecraft to execute a planar docking with a tumbling non-cooperative target; both minimum time and minimum energy maneuvers were considered. The simulation results presented showed that the IDVD–SGRA is capable of computing solution trajectories with CPU time of the order of a second (using a Thinkpad Lenovo T500) with performance index values very close to the correspondent values obtained from the optimal solution.

On a subsequent work [2] the same method has been implemented as guidance strategy for spacecraft translation-only proximity maneuvers in Clohessy–Wiltshire dynamics with path constraints capability. Initially the IDVD–SGRA was exploited to rapid prototype minimum energy trajectories which transfer a spacecraft from a given initial state, described in terms of position and velocity, to another final state while avoiding a keep-out-zone located along the path. On a second phase the IDVD–SGRA has been implemented on a closed-loop fashion; namely, along the trajectory, the spacecraft position components and velocity components are updated at a certain sample frequency and the IDVD–SGRA is reiterated to compute refreshed values of the thrust components. This closed-loop implementation allows compensating unpredicted system disturbances such as: external forces acting on the spacecraft, sensor noise, and uncertainties on the system (mass change, thrust variation, etc.).

Other successful applications of the IDVD involve the generation of 6 DOF minimum time and minimum energy rendezvous trajectories with a tumbling object [3], and 3 DOF translation-only docking trajectory for an orbiting spacecraft [4].

* Corresponding author. Tel.: +1 8316562150.
E-mail addresses: mciarcia@nps.edu, cimarco74@hotmail.com (M. Ciarcià).

0094-5765/$ - see front matter © 2014 IAA. Published by Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.actaastro.2014.01.002

In this work, we exploit experimentally the IDVD–SGRA as near-optimal guidance strategy for the autonomous cooperative planar docking maneuvers between two Floating Spacecraft Simulators (FSS). The Floating Spacecraft-Simulator Testbed [5,6] is an on-ground experimental facility composed of a set of four autonomous FSSs capable of flotation (using airpads) over a high precision flat surface (4 m × 4 m); a set of eight compressed air fed supersonic thrusters provides the thrust and torque required to maneuver over the flat floor.

In order to perform cooperative minimum energy planar docking, the IDVD–SGRA has been implemented in closed loop fashion on the two FSSs. Each FSS will behave as a chaser which must dock with the other FSS (the target). While maneuvering, at every sample time interval, the current position and velocity of chaser and target are updated and the near-optimal trajectory is recomputed to attain refreshed values of the controls to be commanded on the subsequent sample interval. All the near-optimal trajectories computed using IDVD are intrinsically characterized by continuous controls, at the same time the SGRA enforces bounds on the commanded thrust, commanded torque, and maximum maneuver time.

On a second phase we have considered a different cooperative docking maneuver; namely, one FSS will act as a chaser trying to reduce the separation distance from the target which instead will stay at rest in its initial position while aiming its docking interface toward the target.

For comparison purposes, the same set of docking maneuvers has also been executed using an LQR controller [7–11]. The results show that the docking trajectories performed with the IDVD–SGRA guidance are characterized by a lower amount of total maneuver energy and fuel consumption with respect to the LQR.

2. Inverse dynamics in the virtual domain

The planar floating motion of the FSS is described by the following equations (Fig. 1)

$$F_x = m\ddot{x},\quad F_y = m\ddot{y},\quad T_z = I_z\ddot{\theta}, \tag{1}$$

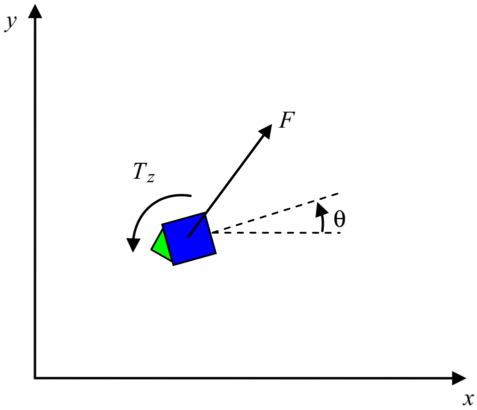

Fig. 1. Floating Spacecraft Simulator reference system.

where m and I_z are the FSS′s mass and moment of inertia, F_x and F_y are the thrust components, and T_z is the torque along the z-axis which is orthogonal to the xy-plane and directed in accordance to the right hand rule.

Let x_0, y_0, θ_0, $\dot{x}_0, \dot{y}_0, \dot{\theta}_0$ andx_f, y_f, θ_f, $\dot{x}_f, \dot{y}_f, \dot{\theta}_f$ be the FSS initial and final states, respectively. The reference optimal trajectory we are interested in is the solution of the following Bolza–Pontryagin optimal control problem: minimize the maneuver energy

$$E = \frac{1}{2}\int_0^{t_f} (F_x^2 + F_y^2 + T_z^2)dt \tag{2}$$

with respect to the controls F_x, F_y, T_z in order to verify the differential constraints in Eq. (1), the inequality constraints

$$|F_x(t)\cos(\theta(t)) + F_y(t)\sin(\theta(t))| \le F_{\max}, \tag{3}$$

$$|-F_x(t)\sin(\theta(t)) + F_y(t)\cos(\theta(t))| \le F_{\max}, \tag{4}$$

$$|T_z(t)| \le T_{z\ \max}, \tag{5}$$

$$t_f \le t_{\max}, \tag{6}$$

and the boundary conditions

$$x(0) = x_0, y(0) = y_0, \theta(0) = \theta_0, x(t_f) = x_f, y(t_f) = y_f, \theta(t_f) = \theta_f, \tag{7}$$

$$\dot{x}(0) = \dot{x}_0, \dot{y}(0) = \dot{y}_0, \dot{\theta}(0) = \dot{\theta}_0, \dot{x}(t_f) = \dot{x}_f, \dot{y}(t_f) = \dot{y}_f, \dot{\theta}(t_f) = \dot{\theta}_f. \tag{8}$$

Eqs. (3) and (4) limit the thrust components in body axis to the maximum thrust available $F_{\max}$, Eq. (5) limits the torque commanded to the maximum torque available $T_{z\ \max}$, and Eq. (6) limits the maneuver time to a predefined maximum value $t_{\max}$.

The goal of the IDVD is to convert such optimal control problem into an equivalent nonlinear programming problem; namely, the problem of minimizing the performance index

$$g = g(r_1, r_2, \ldots, r_k) \tag{9}$$

with respect to a set of varied parameters $[r_1, r_2, \ldots, r_k]$, subject to the constraints

$$\mathbf{h}(r_1, r_2, \ldots, r_k) = 0. \tag{10}$$

The first step is to describe the FSS trajectory components and time with a set of basis functions defined in a virtual variable domain [12,13]. For this particular application, these components (position and attitude) are expressed as a set of polynomials

$$x(\tau) = \sum_{i=0}^{s} a_i\tau^i, y(\tau) = \sum_{i=0}^{s} b_i\tau^i, \theta(\tau) = \sum_{i=0}^{s} c_i\tau^i, \tau \in [0, \tau_f], \tag{11}$$

where τ is an independent virtual argument. The coefficients a_i, b_i, c_i are related to the initial and final conditions on FSS position, velocity, acceleration, etc. In particular, they can be chosen in such a way that these boundary conditions are automatically satisfied. For this purpose, the polynomials degree s is strictly related to the sum of initial and final conditions that need to be imposed; namely, the following inequality must hold [2]

$$s \ge O_0 + O_f + 1, \tag{12}$$

where O_0 and O_f are the time derivative higher orders of the assigned initial and final conditions, respectively. For cooperative docking maneuvers initial and final conditions on both position and velocity must be imposed, that is, $s \geq 3$. The final choice of the polynomial degree is driven by a compromise between the degrees of freedom we want to introduce on the trajectory (in terms of shape complexity) and computational burden. This guidance application requires maximum computational speed performances, for this reason we have chosen the polynomial of the lowest possible order by assuming

$$\begin{aligned} x(\tau) &= a_0 + a_1\tau + a_2\tau^2 + a_3\tau^3 \\ y(\tau) &= b_0 + b_1\tau + b_2\tau^2 + b_3\tau^3 \\ \theta(\tau) &= c_0 + c_1\tau + c_2\tau^2 + c_3\tau^3, \tau \in [0, \tau_f]. \end{aligned} \tag{13}$$

As regards to the mapping between the time domain and the virtual argument domain, it is also expressed through the polynomial expression

$$t = \sum_{i=1}^{p} d_i \tau^i, \tau \in [0, \tau_f]. \tag{14}$$

In this case, the polynomial order p affects the flexibility on the speed changes along the trajectory and computational burden as well. Previous investigations [1,2] have shown that a good compromise for planar maneuvers is $p=2$

$$t = d_1\tau + d_2\tau^2, \tau \in [0, \tau_f]. \tag{15}$$

Note that an analytical description of the trajectory is attainable once the coefficients a_i, b_i, c_i, d_i and the final value of the virtual variable τ_f are known.

2.1. Boundary conditions and dynamics inversion

As mentioned before, the coefficients a_i, b_i, c_i are related to the initial and final conditions on position and velocity. From Eq. (7) we have

$$x(0) = a_0 = x_0, x(\tau_f) = a_0 + a_1\tau_f + a_2\tau_f^2 + a_3\tau_f^3 = x_f. \tag{16}$$

The boundary conditions on velocity components and angular rate, Eq. (8), cannot be directly imposed without some introductory considerations. Let the symbol prime "′" denote the operator derivative with respect to the virtual variable τ. It results:

$$x' = \frac{dx}{d\tau} = \frac{dx}{dt}\frac{dt}{d\tau} = \dot{x}\frac{dt}{d\tau}, \tag{17}$$

where

$$x' = a_1 + 2a_2\tau + 3a_3\tau^2, \tag{18}$$

$$\frac{dt}{d\tau} = d_1 + 2d_2\tau. \tag{19}$$

The boundary conditions on velocity components then become

$$x'(0) = a_1 = \dot{x}_0 d_1, x'(\tau_f) = a_1 + 2a_2\tau_f + 3a_3\tau_f^2 = \dot{x}_f(d_1 + 2d_2\tau_f). \tag{20}$$

Solving for a_0, a_1, a_2, a_3 we obtain

$$\begin{aligned} a_0 &= x_0, a_1 = \dot{x}_0 d_1, a_2 = \frac{-3x_0 - 2\dot{x}_0\tau_f d_1 + 3x_f - \dot{x}_f\tau_f(d_1 + 2d_2\tau_f)}{\tau_f^2}, \\ a_3 &= \frac{2x_0 + \dot{x}_0\tau_f d_1 - 2x_f + \dot{x}_f\tau_f(d_1 + 2d_2\tau_f)}{\tau_f^3}. \end{aligned} \tag{21}$$

Similarly, by imposing the boundary conditions on the y-axis $y_0, y_f, \dot{y}_0, \dot{y}_f$ and rotational axis $\theta_0, \theta_f, \dot{\theta}_0, \dot{\theta}_f$, we obtain the following expressions for the coefficients b_0, b_1, b_2, b_3, c_0, c_1, c_2, c_3

$$\begin{aligned} b_0 &= y_0, b_1 = \dot{y}_0 d_1, b_2 = \frac{-3y_0 - 2\dot{y}_0\tau_f d_1 + 3y_f - \dot{y}_f\tau_f(d_1 + 2d_2\tau_f)}{\tau_f^2}, \\ b_3 &= \frac{2y_0 + \dot{y}_0\tau_f d_1 - 2y_f + \dot{y}_f\tau_f(d_1 + 2d_2\tau_f)}{\tau_f^3}, \end{aligned} \tag{22}$$

$$\begin{aligned} c_0 &= \theta_0, c_1 = \dot{\theta}_0 d_1, c_2 = \frac{-3\theta_0 - 2\dot{\theta}_0\tau_f d_1 + 3\theta_f - \dot{\theta}_f\tau_f(d_1 + 2d_2\tau_f)}{\tau_f^2}, \\ c_3 &= \frac{2\theta_0 + \dot{\theta}_0\tau_f d_1 - 2\theta_f + \dot{\theta}_f\tau_f(d_1 + 2d_2\tau_f)}{\tau_f^3}. \end{aligned} \tag{23}$$

By substituting these expressions in Eq. (13) we achieve the following analytical descriptions of the position components and attitude in the virtual domain

$$\begin{aligned} x &= x(x_0, x_f, \dot{x}_0, \dot{x}_f, d_1, d_2, \tau_f, \tau), \\ y &= y(y_0, y_f, \dot{y}_0, \dot{y}_f, d_1, d_2, \tau_f, \tau), \\ \theta &= \theta(\theta_0, \theta_f, \dot{\theta}_0, \dot{\theta}_f, d_1, d_2, \tau_f, \tau), \quad \tau \in [0, \tau_f]. \end{aligned} \tag{24}$$

Exploiting the derivative rule in Eq. (17), we obtain a similar expression for the velocity components and angular rate

$$\begin{aligned} \dot{x} &= \dot{x}(x_0, x_f, \dot{x}_0, \dot{x}_f, d_1, d_2, \tau_f, \tau), \\ \dot{y} &= \dot{y}(y_0, y_f, \dot{y}_0, \dot{y}_f, d_1, d_2, \tau_f, \tau), \\ \dot{\theta} &= \dot{\theta}(\theta_0, \theta_f, \dot{\theta}_0, \dot{\theta}_f, d_1, d_2, \tau_f, \tau), \quad \tau \in [0, \tau_f]. \end{aligned} \tag{25}$$

As regard to the acceleration component $\ddot{x}$, from Eq. (17) it results

$$\ddot{x} = \frac{d\dot{x}}{dt} = \frac{d}{dt}\left(\frac{x'}{dt/d\tau}\right) = \frac{d\tau}{dt}\frac{d}{d\tau}\left(\frac{x'}{dt/d\tau}\right) = \frac{1}{(dt/d\tau)^2}\left(x'' - \dot{x}\frac{d^2t}{d\tau^2}\right). \tag{26}$$

Similarly, for $\ddot{y}$ and $\ddot{\theta}$ we get

$$\ddot{y} = \frac{d\dot{y}}{dt} = \frac{1}{(dt/d\tau)^2}\left(y'' - \dot{y}\frac{d^2t}{d\tau^2}\right), \tag{27}$$

$$\ddot{\theta} = \frac{d\dot{\theta}}{dt} = \frac{1}{(dt/d\tau)^2}\left(\theta'' - \dot{\theta}\frac{d^2t}{d\tau^2}\right). \tag{28}$$

This procedure allows us to attain analytical expressions in the virtual domain also for the acceleration components and angular acceleration

$$\begin{aligned} \ddot{x} &= \ddot{x}(x_0, x_f, \dot{x}_0, \dot{x}_f, d_1, d_2, \tau_f, \tau), \\ \ddot{y} &= \ddot{y}(y_0, y_f, \dot{y}_0, \dot{y}_f, d_1, d_2, \tau_f, \tau), \\ \ddot{\theta} &= \ddot{\theta}(\theta_0, \theta_f, \dot{\theta}_0, \dot{\theta}_f, d_1, d_2, \tau_f, \tau), \quad \tau \in [0, \tau_f]. \end{aligned} \tag{29}$$

The main result of this process is that, through the inversion of the FSS dynamics

$$F_x = m\ddot{x},$$

$$F_y = m\ddot{y},$$
$$T_z = I_z\ddot{\theta}, \tag{30}$$

We have obtained the analytical expressions in the virtual domain of the controls acting on the FSS

$$\begin{aligned}
F_x &= F_x(m, x_0, x_f, \dot{x}_0, \dot{x}_f, d_1, d_2, \tau_f, \tau),\\
F_y &= F_y(m, y_0, y_f, \dot{y}_0, \dot{y}_f, d_1, d_2, \tau_f, \tau),\\
T_z &= T_z(I_z, \theta_0, \theta_f, \dot{\theta}_0, \dot{\theta}_f, d_1, d_2, \tau_f, \tau), \quad \tau \in [0, \tau_f].
\end{aligned} \tag{31}$$

These expressions involve a set of given parameters $m, I_z, x_0, y_0, \theta_0, \dot{x}_0, \dot{y}_0, \dot{\theta}_0,\ x_f, y_f, \theta_f, \dot{x}_f, \dot{y}_f, \dot{\theta}_f$ a set of varied parameters d_1, d_2, τ_f and the virtual argument τ.

2.2. Nonlinear programming problem

As mentioned before, for the cooperative planar docking, the optimization criterion chosen is the maneuver energy

$$E = \frac{1}{2}\int_0^{t_f}(F_x^2 + F_y^2 + T_z^2)dt. \tag{32}$$

In virtue of Eq. (17) and exploiting Eq. (31) we can convert this integral in the time domain into an integral in the virtual domain

$$E = \frac{1}{2}\int_0^{\tau_f}(F_x^2 + F_y^2 + T_z^2)\frac{dt}{d\tau}d\tau = \frac{1}{2}\int_0^{\tau_f}(F_x^2 + F_y^2 + T_z^2)(d_1 + 2d_2\tau)d\tau, \tag{33}$$

which implies

$$E = E(m, I_z, x_0, y_0, \theta_0, \dot{x}_0, \dot{y}_0, \dot{\theta}_0, x_f, y_f, \theta_f, \dot{x}_f, \dot{y}_f, \dot{\theta}_f, d_1, d_2, \tau_f). \tag{34}$$

As regard to the bounds on the controls in Eqs. (3)–(5), the eight microthrusters on board of the FSS can produce thrust components and torque subject to the following constraints

$$\begin{aligned}
&|F_{x_B}(t)| \le F_{\max},\\
&|F_{y_B}(t)| \le F_{\max},\\
&|T_z(t)| \le T_{z\ \max}, \quad t \in [0, t_f]
\end{aligned} \tag{35}$$

where $F_{x_B}(t)$ and $F_{y_B}(t)$ are the thrust components projected in body axis

$$\begin{aligned}
F_{x_B}(t) &= F_x(t)\cos(\theta(t)) + F_y(t)\sin(\theta(t)),\\
F_{y_B}(t) &= -F_x(t)\sin(\theta(t)) + F_y(t)\cos(\theta(t)).
\end{aligned} \tag{36}$$

Let us discretize the virtual domain $[0,\tau_f]$ in $m+1$ equidistant nodes $0 = \tau_0 < \tau_1 < \dots < \tau_m = \tau_f$ such that

$$\tau_j = \frac{j}{m}\tau_f\ , \quad j \in \{0, 1, 2, \dots, m\}, \tag{37}$$

to which correspond, in the time domain, the nodes $0 = t_0 < t_1 < \dots < t_m = t_f$

$$t_j = d_1\tau_j + d_2\tau_j^2, \quad j \in \{0, 1, 2, \dots, m\}. \tag{38}$$

An indirect enforcement of the inequalities in Eq. (35) is obtained introducing the slack variables e_{Fx}, e_{Fy}, e_{Tz} and imposing the equalities

$$\begin{aligned}
&\max((F_x(\tau_j)\cos(\theta(\tau_j)) + F_y(\tau_j)\sin(\theta(\tau_j)))^2) + e_{Fx}^2 = F_{\max}^2,\\
&\max((-F_x(\tau_j)\sin(\theta(\tau_j)) + F_y(\tau_j)\cos(\theta(\tau_j)))^2) + e_{Fy}^2 = F_{\max}^2,\\
&\max((T_z(\tau_j))^2) + e_{Tz}^2 = T_{z\ \max}^2, \quad j \in \{0, 1, 2, \dots, m\}.
\end{aligned} \tag{39}$$

It is very important to note that the maxima in Eq. (39) happen to be enforced in correspondence of the virtual time nodes that are fractions of τ_f (see Eq. (37)). In this way, it is possible to relate those equality conditions to this varied parameter. Needless to mention, this method does not provide a rigorous enforcement of the constraints, in fact the inequalities in Eq. (29) will be guaranteed only in proximity of the actual maxima. Obviously, higher number of nodes will imply more accurate enforcement of the bounds but with the disadvantage of a higher number of numerical evaluations. For practical applications, values of m ranging between 20 and 100 provide, usually, satisfactory results.

The inequality constraint on the maneuver time

$$t_f \le t_{\max}, \tag{40}$$

can be enforced by introducing the slack variable e_t and the equality

$$(d_1\tau_f + d_2\tau_f^2)^2 + e_t^2 = t_{\max}^2. \tag{41}$$

We can finally describe our nonlinear programming problem in aforementioned form, that is: minimize the performance index

$$g = g(r_1, r_2, \dots, r_k) \tag{42}$$

with respect to a set of varied parameters $[r_1,r_2,\dots,r_k]$, subject to the constraints

$$\mathbf{h}(r_1, r_2, \dots, r_k) = 0. \tag{43}$$

In this case, the problem becomes: given the set of given parameters $m, I_z, x_0, y_0, \theta_0,\ \dot{x}_0, \dot{y}_0, \dot{\theta}_0, x_f, y_f, \theta_f, \dot{x}_f, \dot{y}_f, \dot{\theta}_f$ minimize the performance index

$$E = E(m, I_z, x_0, y_0, \theta_0, \dot{x}_0, \dot{y}_0, \dot{\theta}_0, x_f, y_f, \theta_f, \dot{x}_f, \dot{y}_f, \dot{\theta}_f, d_1, d_2, \tau_f) \tag{44}$$

with respect to the varied parameters $d_1, d_2, \tau_f, e_{Fx}, e_{Fy}, e_{Tz}, e_t$ subject to the constraints

$$\begin{aligned}
&\max((F_x(\tau_j)\cos(\theta(\tau_j)) + F_y(\tau_j)\sin(\theta(\tau_j)))^2) + e_{Fx}^2 = F_{\max}^2,\\
&\max((-F_x(\tau_j)\sin(\theta(\tau_j)) + F_y(\tau_j)\cos(\theta(\tau_j)))^2) + e_{Fy}^2 = F_{\max}^2,\\
&\max((T_z(\tau_j))^2) + e_{Tz}^2 = T_{z\ \max}^2, \quad j \in \{0, 1, 2, \dots, m\},\\
&(d_1\tau_f + d_2\tau_f^2)^2 + e_t^2 = t_{\max}^2.
\end{aligned} \tag{45}$$

3. Sequential gradient-restoration algorithm

The algorithm chosen to solve the previous optimization problem is the nonlinear programming version of the SGRA [14]. SGRA is based on two scalar quantities; the constraint error $P = \mathbf{h}^T\mathbf{h}$ and the optimality condition error $Q = {G_r}^T G_r$ (see Eqs. (9) and (10)), where

$$G = g + \boldsymbol{\eta}^T\mathbf{h} \tag{46}$$

is the augmented function associated to the performance index g and the constraints function $\mathbf{h}$, and

$$G_r = \left(\frac{\partial G}{\partial r_1}, \frac{\partial G}{\partial r_2}, \dots, \frac{\partial G}{\partial r_k}\right)^T \tag{47}$$

is the gradient of the augmented function with respect to vector of varied parameters $[r_1,r_2,\dots,r_k]$. In Eq. (46), $\boldsymbol{\eta}$ is the

Lagrange multiplier vector associated with the constraints function **h**.

SGRA involves a cyclical scheme whereby first the constraints are satisfied to a prescribed accuracy (restoration phase); then, using a first-order gradient method, a step is taken toward the optimal direction to improve the performance index (gradient phase). Note that, in this implementation, Eqs. (35) and (40) are satisfied at the end of each restoration phase. The convergence tolerances for each phase are expressed as follows:

$$P \le \varepsilon_1 \tag{48}$$

for the restoration phase and

$$Q \le \varepsilon_2 \tag{49}$$

for the gradient phase, with ε_1 and ε_2 small preselected positive constants.

The qualifier "sequential" is most appropriate for this algorithm due to the successive application of either a gradient iteration or a restorative iteration depending on the values of the scalar quantities P and Q. Specifically, three cases can occur:

(C1) If $P > \varepsilon_1$, SGRA executes a restorative iteration leading to the decrease of the constraint error.
(C2) If $P \le \varepsilon_1$ but $Q > \varepsilon_2$, SGRA executes a gradient iteration leading to the decrease of the augmented function.
(C3) If $P \le \varepsilon_1$ and $Q \le \varepsilon_2$, convergence is declared and the algorithm stops.

In general, achieving specified final conditions in a feasible way is more important than obtaining an exact optimal solution: so, the constraint requirements (restoration phase) are stricter than the optimality condition requirements (gradient phase), that is $\varepsilon_1 \ll \varepsilon_2$.

4. Cooperative docking trajectories

The IDVD–SGRA method computes near-optimal trajectories which transfer the FSS from some given initial position and velocity to some given final position and velocity.

Let's indicate with FSS A and FSS B the two vehicles and with x_{A_0}, y_{A_0} and x_{B_0}, y_{B_0} their center of mass initial positions, respectively. At the beginning of the maneuver both are at rest with zero initial attitude and the docking interfaces oriented toward the negative direction of the x-axis as shown in Fig. 2. The initial conditions, for the two FSSs, are then the following:

FSS A

$$\begin{aligned} &x_A(0) = x_{A_0}, y_A(0) = y_{A_0}, \theta_A(0) = 0,\\ &\dot{x}_A(0) = 0, \dot{y}_A(0) = 0, \dot{\theta}_A(0) = 0. \end{aligned} \tag{50}$$

FSS B

$$\begin{aligned} &x_B(0) = x_{B_0}, y_B(0) = y_{B_0}, \theta_A(0) = 0,\\ &\dot{x}_B(0) = 0, \dot{y}_B(0) = 0, \dot{\theta}_B(0) = 0. \end{aligned} \tag{51}$$

As mentioned before, two different type docking executions have been investigated, this implies that two different set of final conditions must be imposed:

($D1$) During the maneuver both FSSs behave as chasers. In this case the common goal is to reduce the separation distance from the other FSS and to complete the maneuver with the matching final attitude. In order to achieve the docking configuration it is necessary that the following final conditions are satisfied:

FSS A

$$\begin{aligned} &x_A(t_f) = x_{B_f} - d\ \cos(\theta_{B_f} + \pi),\\ &y_A(t_f) = y_{B_f} - d\ \sin(\theta_{B_f} + \pi),\\ &\theta_A(t_f) = \theta_{B_f} + \pi,\\ &\dot{x}_A(t_f) = 0, \dot{y}_A(t_f) = 0, \dot{\theta}_A(t_f) = 0. \end{aligned} \tag{52}$$

FSS B

$$\begin{aligned} &x_B(t_f) = x_A(t_f) - d\ \cos(\theta_B(t_f)),\\ &y_B(t_f) = y_A(t_f) - d\ \sin(\theta_B(t_f)),\\ &\theta_B(t_f) = \theta_A(t_f) + \pi,\\ &\dot{x}_B(t_f) = 0, \dot{y}_B(t_f) = 0, \dot{\theta}_B(t_f) = 0, \end{aligned} \tag{53}$$

where d is the distance between the two FSS's center of mass in docked configuration (Fig. 2). The need to add π

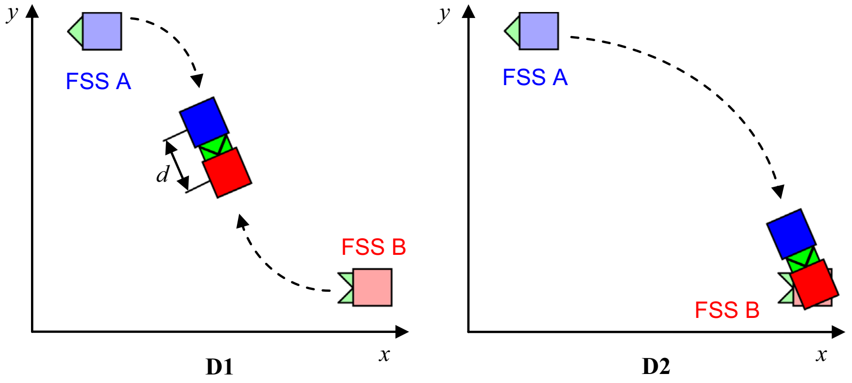

Fig. 2. Planar docking execution schemes.

in the expressions of $\theta_{\mathrm{A}}(t_f)$ and $\theta_{\mathrm{B}}(t_f)$ is due to the fact that the docking interface is located, in both vehicles, on the opposite side from where the attitude is measured (Fig. 1).

($D2$) In this case only FSS A behaves as chaser trying to minimize its distance from FSS B which instead will keep its initial position while aiming its docking interface toward the chaser. In other words only one of the two vehicles is in charge to null the separation distance whereas the other will just wait and modifying its attitude to match the final chaser attitude. In this scenario the final conditions become

FSS A

$$\begin{aligned}
&x_{\mathrm{A}}(t_f) = x_{\mathrm{B}_f} - d\ \cos(\theta_{\mathrm{B}_f} + \pi),\\
&y_{\mathrm{A}}(t_f) = y_{\mathrm{B}_f} - d\ \sin(\theta_{\mathrm{B}_f} + \pi),\\
&\theta_{\mathrm{A}}(t_f) = \theta_{\mathrm{B}_f} + \pi,\\
&\dot{x}_{\mathrm{A}}(t_f) = 0, \dot{y}_{\mathrm{A}}(t_f) = 0, \dot{\theta}_{\mathrm{A}}(t_f) = 0.
\end{aligned} \tag{54}$$

FSS B

$$\begin{aligned}
&x_{\mathrm{B}}(t_f) = x_{\mathrm{B}_0},\ y_{\mathrm{B}}(t_f) = y_{\mathrm{B}_0}, \theta_{\mathrm{B}}(t_f) = \theta_{\mathrm{A}_f} + \pi,\\
&\dot{x}_{\mathrm{B}}(t_f) = 0, \dot{y}_{\mathrm{B}}(t_f) = 0, \dot{\theta}_{\mathrm{B}}(t_f) = 0.
\end{aligned} \tag{55}$$

It is important to note that the initial position components $x_{\mathrm{A}_0}, y_{\mathrm{A}_0}, x_{\mathrm{B}_0}, y_{\mathrm{B}_0}$ are given parameters whereas the final position components and attitude $x_{\mathrm{A}_f}, y_{\mathrm{A}_f}, \theta_{\mathrm{A}_f}, x_{\mathrm{B}_f}, y_{\mathrm{B}_f}, \theta_{\mathrm{B}_f}$ are unknown parameters. As we will explain better in the following section, the closed loop implementation of the IDVD–SGRA will allow us to make an estimate of these last set of parameters during the docking execution. Such estimation will become more accurate toward the final part of the maneuver and becomes a precise guess by the end of it.

4.1. IDVD–SGRA closed-loop implementation

In a previous paper [1] we have tested the capability of the IDVD–SGRA method to fast prototype near-optimal trajectories. In detail, for the IDVD–SGRA with 3rd order polynomial description of position coordinates and attitude, it can converge to the solution with CPUtime of the order of hundredths of a second. This performance makes it suitable for closed-loop implementation in a decreasing horizon fashion. During the maneuver, at each sample time interval (0.5 s), the current position components and velocity components of chaser and target are updated, then a new near-optimal trajectory is recomputed and refreshed values of force components and torque are commanded (Fig. 3). This process is reiterated until completion of the maneuver. The remaining maximum maneuver time, which at the beginning of the maneuver is equal to t_{MAX}, will decrease progressively while the maneuver time is elapsing.

In this scenario it becomes clear that, at each sample time interval, the initial conditions to be updated in Eq. (48), for FSS A, are the following:

$$\begin{aligned}
&x_{\mathrm{A}_0} = x_C(t_i), y_{\mathrm{A}_0} = y_C(t_i), \theta_{\mathrm{A}_0} = \theta_C(t_i),\\
&\dot{x}_{\mathrm{A}_0} = \dot{x}_C(t_i), \dot{y}_{\mathrm{A}_0} = \dot{y}_C(t_i), \dot{\theta}_{\mathrm{A}_0} = \dot{\theta}_C(t_i).
\end{aligned} \tag{56}$$

As regard to the final conditions to be updated in Eqs. (50) and (52), from the current status of the target $x_T(t_i), y_T(t_i), \theta_T(t_i), \dot{x}_T(t_i), \dot{y}_T(t_i), \dot{\theta}_T(t_i)$,we can estimate the final docking configuration of the target (FSS B) by assuming the following linear motion propagation

$$\begin{aligned}
&x_{\mathrm{B}_f}(t_i) = x_T(t_i) + K\dot{x}_T(t_i)(t_{\mathrm{MAX}} - t_i),\\
&y_{\mathrm{B}_f}(t_i) = y_T(t_i) + K\dot{y}_T(t_i)(t_{\mathrm{MAX}} - t_i),
\end{aligned} \tag{57}$$

where K is a nonnegative correction constant which we have introduced to take into account the changeable behavior of the target. During our experimental campaign, we have found heuristically that $K = 0.6$ produces satisfactory results. Note that for $t_i \to t_{\mathrm{MAX}}$ we have $x_{\mathrm{B}f}\ (t_i) \to x_T\ (t_i)$ and $y_{\mathrm{B}f}\ (t_i) \to y_T\ (t_i)$, this confirm that at the end of the maneuver the target estimated position coincides with its actual position. An identical approach has been used to update FSS B′s initial and final conditions.

As for the desired final attitude $\theta_{\mathrm{B}f}$, it has been defined as the angle between the x-axis and the line which goes from the center of mass of FSS A to the center of mass of FSS B, i.e. if

$$\begin{aligned}
&\Delta x = x_{\mathrm{B}_f}(t_i) - x_{\mathrm{A}_f}(t_i),\\
&\Delta y = y_{\mathrm{B}_f}(t_i) - y_{\mathrm{A}_f}(t_i),\\
&\alpha = \arctan\left(\frac{\Delta y}{\Delta x}\right)
\end{aligned} \tag{58}$$

it results

$$\theta_{\mathrm{B}_f}\left(t_i\right) = \begin{cases} \alpha & \Delta x > 0, \Delta y > 0 \text{ or } \quad \Delta x > 0, \Delta y < 0 \\ \alpha + \pi & \Delta x < 0, \qquad\qquad\qquad \Delta y > 0 \\ \alpha - \pi & \Delta x < 0, \qquad\qquad\qquad \Delta y < 0 \\ \frac{\pi}{2} & \Delta x = 0, \qquad\qquad\qquad \Delta y \geq 0 \\ -\frac{\pi}{2} & \Delta x = 0, \qquad\qquad\qquad \Delta y < 0 \end{cases} \tag{59}$$

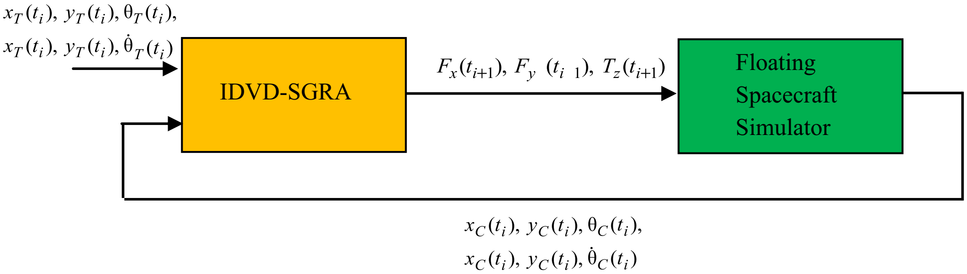
$x_T(t_i)$, $y_T(t_i)$, $\theta_T(t_i)$,
$\dot{x}_T(t_i)$, $\dot{y}_T(t_i)$, $\dot{\theta}_T(t_i)$
IDVD-SGRA
$F_x(t_{i+1})$, $F_y\ (t_i\ _1)$, $T_z(t_{i+1})$
Floating Spacecraft Simulator
$x_C(t_i)$, $y_C(t_i)$, $\theta_C(t_i)$,
$\dot{x}_C(t_i)$, $\dot{y}_C(t_i)$, $\dot{\theta}_C(t_i)$

Fig. 3. IDVD–SGRA closed-loop guidance scheme.

A similar definition has been exploited to define the desired final attitude θ_{Af}.

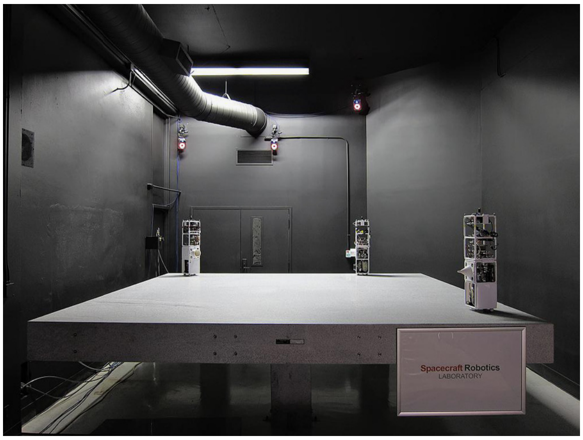

Fig. 4. The Floating Spacecraft-Simulator Testbed at Spacecraft Robotics Laboratory.

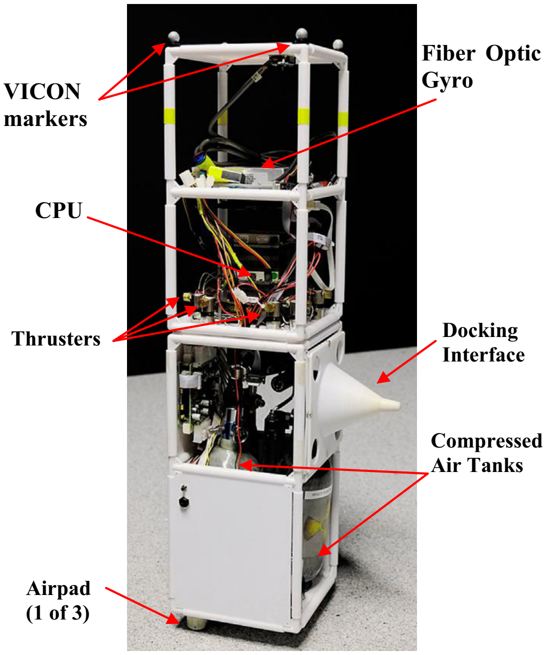

Fig. 5. Floating Spacecraft Simulator.

5. Floating Spacecraft-Simulator Testbed

Our experimental campaign has been conducted at the recently upgraded Floating Spacecraft-Simulator Testbed [5,6] (Fig. 4). This unique on-ground experimental facility is composed of a set of four FSSs and a high precision floating surface located in a clean/low reflective environment. The dynamics of the FSS on the flat surface reproduce closely, in 2D, the weightlessness and frictionless conditions of the relative 3D orbital flight.

The main floating surface is a granite monolith with the following characteristics:

- Dimensions: $4\ \text{m} \times 4\ \text{m} \times 0.3\ \text{m}$
- Surface precision grade: AAA
- Planar accuracy: $\pm 0.127 \times 10^{-2}$ mm
- Horizontal leveling precision: < 0.01 deg
- Mass: 15.2×10^3 kg.

The testbed is also provided with:

1. VICON tracking system (10 cameras, tracking accuracy lower than 1 mm for the entire volume of the laboratory)
2. Linux Real-Time work station
3. Ad-Hoc WiFi internal network for data streaming
4. High pressure air compressor and compressed air filling station.

5.1. Floating Spacecraft Simulator

The Floating Spacecraft Simulator is (Fig. 5) a custom designed autonomous vehicle capable of floatation via airpads and actuated by a set of eight supersonic microthrusters [15] (two for each side) fed with compressed air. Each vehicle is provided with an on-board computer running a real-time operating system, a fiber optic gyro for attitude measurements, and two compressed air tanks which feed the airpads and microthrusters. The two Floating Spacecraft Simulators are characterized by the following specifications:

$$m_A = 9.48\ \text{kg},$$
$$m_B = 9.91\ \text{kg},$$
$$I = 6.02 \times 10^{-2}\ \text{kg m}^2,$$
$$I = 6.14 \times 10^{-2}\ \text{kg m}^2,$$
$$d_{doc} = 0.34\ \text{m},$$
$$F_{max} = 0.12\ N,$$
$$T_{z\ max} = 1.44 \times 10^{-2}\ \text{N m}.$$

Table 1
Test cases.

Case	x_{A_0} (m)	y_{A_0} (m)	x_{B_0} (m)	y_{B_0} (m)	m_{A_0} (kg)	m_{B_0} (kg)	t_{MAX} (s)
1	0.29	3.74	3.68	0.35	9.46	9.89	97
2	1.40	3.69	2.74	0.29	9.42	9.84	80
3	0.41	3.60	3.77	1.85	9.42	9.84	84
4	1.44	3.57	3.38	1.34	9.43	9.83	79

6. Test cases

Four test cases have been considered in our experiments. In Table 1 are shown the initial positions and the initial masses of the two FSSs. To evaluate the IDVD–SGRA performances, for both types docking execution *D*1 and *D*2, the same maneuvers have been performed using an LQR controller [7–11]. Since such controller does not enforce any constraint on the maximum maneuver time, in order to ensure that this parameter is equal for all the controllers, we have first executed the dockings with the LQR then we have set the resulting maneuver times as t_{MAX} for the IDVD–SGRA (Table 1).

7. Results

In Table 2 are shown the global results in terms of total maneuver energy E_{TOT} and propellant mass $m_{Prop\ TOT}$. Those quantities refer to the maneuver energy and propellant mass, respectively, required by both vehicles. For comparison purposes, we will consider as reference the total energy and propellant mass required by the LQR controller.

These results show that the total maneuver energy required by the IDVD–SGRA guidance strategy is about one-sixth of the energy required by the LQR. In terms of fuel consumption, the IDVD–SGRA was able to complete most of the dockings maneuvers using less than one-third of propellant mass consumed by the LQR. From Table 2 we observe that a maneuver characterized by a higher amount of control energy, with respect to another, may not correspond to an increase in propellant mass (e.g. Case 2, IDVD–SGRA D1 and IDVD–SGRA D2). This is due to the intrinsic nature of the Pulse Width Modulator which is used to convert the continuous commanded controls into equivalent sequence of fixed-thrust impulses. In fact, very low values of commanded thrust or torque, which provide positive contribution to the maneuver energy, may not be representable as a sequence of impulses due to the physical limitation of the actuators, namely the inertia of solenoid valves. This implies that those contribute to the computed control energy do no contribute to the fuel mass expenditure.

In Figs. 6–9 are represented the bird′s-eye views of the docking maneuvers for all the cases. These plots show that trajectories performed with the LQR guidance follow a path which connects smoothly the initial positions. The alignment of the docking interfaces is achieved and maintained for a long portion of time prior to the end of the maneuver. This continuous control of the trajectory requires an action on the thrusters which is expensive in terms of fuel consumption. Conversely, the IDVD–SGRA guidance executes the maneuver so that the docking

Table 2
Global results. Maneuver energy and fuel consumption.

Case	Control strategy	E_{TOT} (N^2s)	E_{TOT} (%)	$m_{Prop\ TOT}$ (g)	$m_{Prop\ TOT}$ (%)
1	**LQR**	2.44	100	75	100
	IDVD–SGRA D1	0.43	18	31	41
	IDVD–SGRA D2	0.17	7	17	23
2	**LQR**	2.29	100	71	100
	IDVD–SGRA D1	0.33	14	19	20
	IDVD–SGRA D2	0.29	13	22	30
3	**LQR**	2.51	100	76	100
	IDVD–SGRA D1	0.25	10	14	18
	IDVD–SGRA D2	0.28	11	8	11
4	**LQR**	2.34	100	70	100
	IDVD–SGRA D1	0.23	10	18	26
	IDVD–SGRA D2	0.34	14	10	14

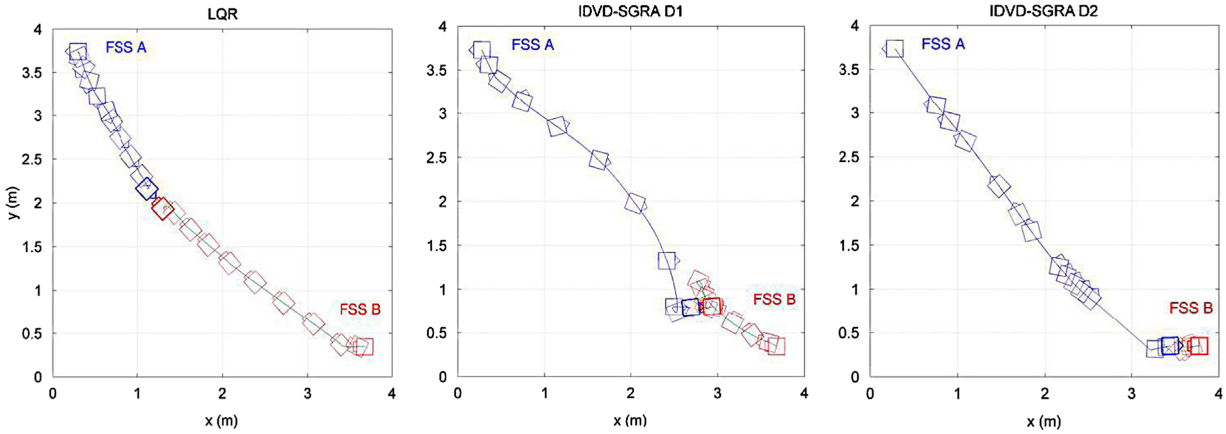

Fig. 6. Case 1. Bird′s-eye views.

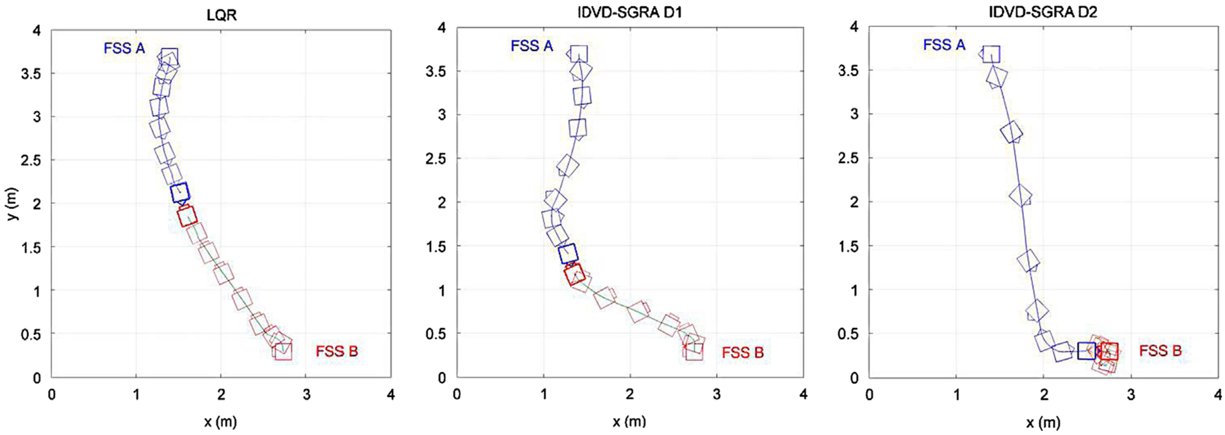

Fig. 7. A Case 2. Bird's-eye views.

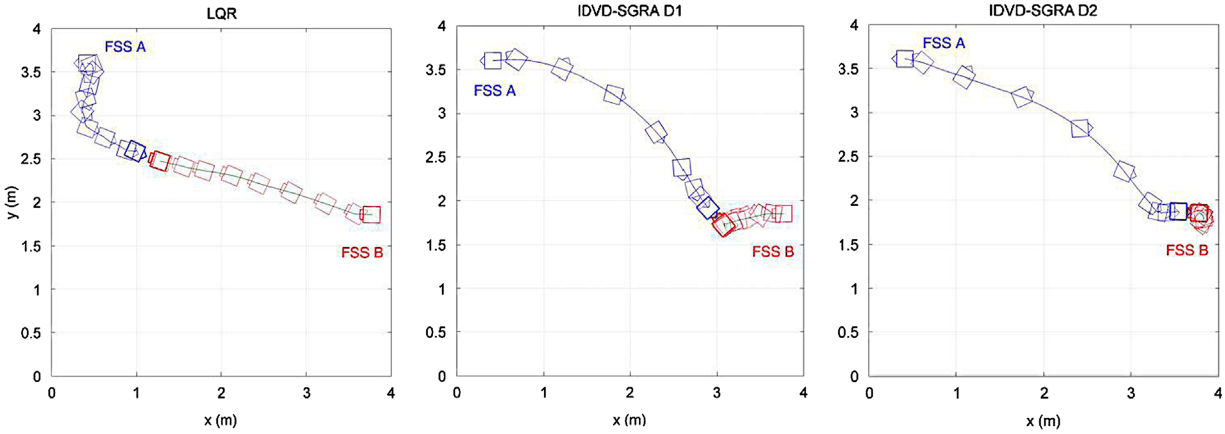

Fig. 8. Case 3. Bird's-eye views.

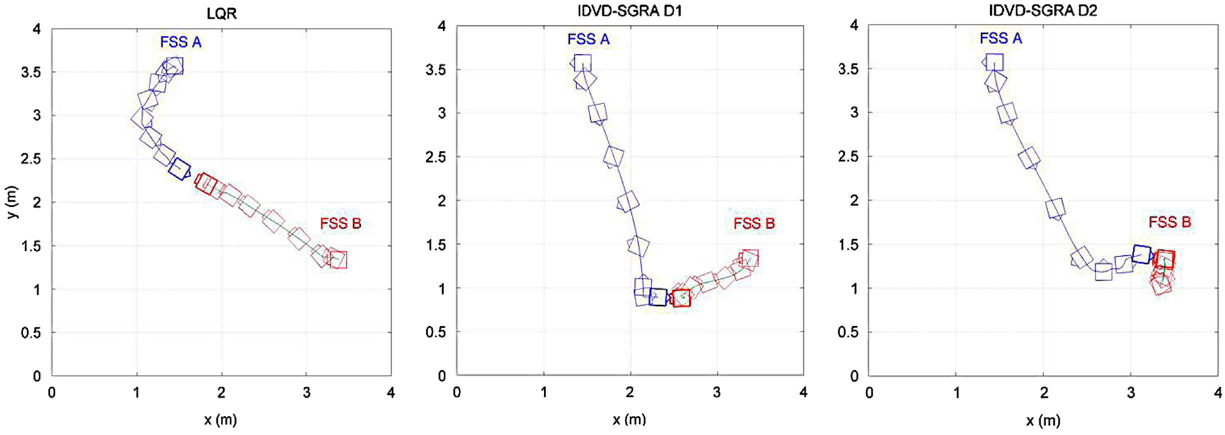

Fig. 9. Case 4. Bird's-eye views.

configuration is achieved only at the end of it. This strategy implies a more efficient bang-off-bang type of behavior on the control time history. Limited actions on the controls are commanded only when consistent trajectory corrections are required. These differences are confirmed by looking at the thruster usage shown in Figs. 10–12. Those refer to the FSS A's thrusters impulses time histories obtained for all the three docking strategies for Case 4. During the last thirty seconds of the maneuver the LQR guidance expends a considerable action of the thrusters (see Fig. 10, whereas IDVD–SGRA D1 and IDVD–SGRA D2 require minimum control.

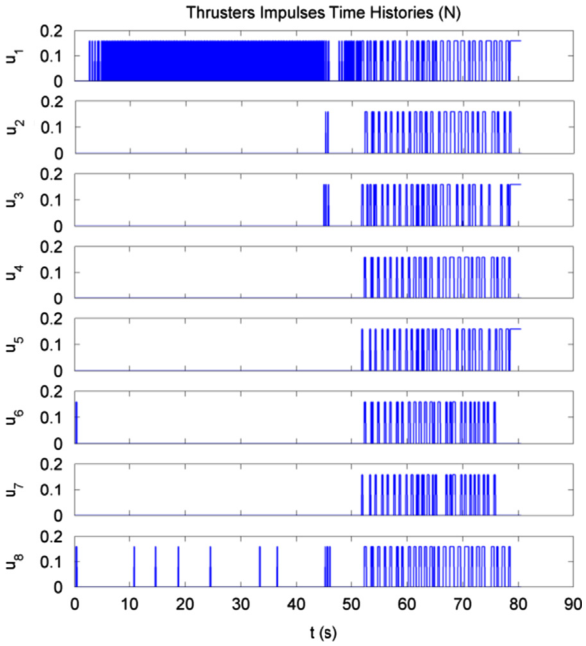

Fig. 10. Case 4. LQR docking. FSS A thrusters impulses time histories.

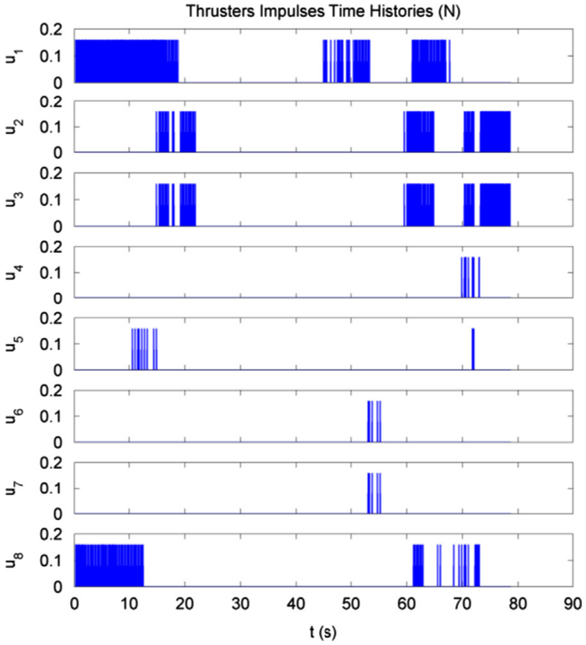

Fig. 11. Case 4. IDVD–SGRA D1 docking. FSS A thrusters impulses time histories.

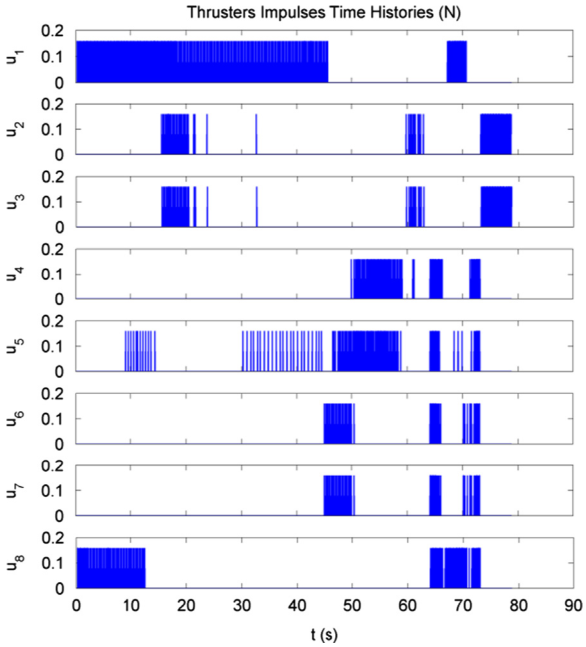

Fig. 12. Case 4. IDVD–SGRA D2 docking. FSS A thrusters impulses time histories.

8. Conclusion

In previous works we have successfully applied the IDVD–SGRA to rapid prototype near-optimal trajectories and as near-optimal guidance for spacecraft proximity maneuvers. The IDVD is a direct method capable of converting an optimal control problem into a nonlinear programming problem, whereas the SGRA is a first-order solver capable of robust and quick convergence performances. In this paper we describe the experimental validation of the IDVD–SGRA as near-optimal guidance strategy for cooperative minimum energy planar docking maneuvers. A set of docking scenarios has been designed and experimented at the Floating Spacecraft-Simulator Testbed. One docking scenario requires each Floating Spacecraft Simulator to behave as a chaser which must dock with the other target vehicle. Subsequently, we have designed a docking scenario in which only one Floating Spacecraft Simulator behaves as a chaser reducing the distance from the target which instead remains in its initial position while aiming its docking interface toward the chaser. For comparison purposes, the same set of docking maneuvers has been performed also using a different controller such as a Linear Quadratic Regulator. The presented results show that the closed-loop implementation of the IDVD–SGRA is capable of executing planar docking maneuvers with a considerable lower amount of energy on the controls and propellant mass.

References

[1] M. Ciarcià, M. Romano, Spacecraft proximity maneuver guidance based on inverse dynamic and sequential gradient-restoration algorithm, in: Proceedings of the AIAA Astrodynamics Specialist Conference, 2011.

[2] M. Ciarcià, M. Romano, Suboptimal guidance for orbital proximity maneuver with path constraints capability, in: Proceedings of the AIAA Guidance Navigation and Control Conference, 2012.

[3] G.A. Boyarko, O.A. Yakimenko, M. Romano, Optimal rendezvous trajectories of a controlled spacecraft and a tumbling object., AIAA J. Guid. Control Dynam 34 (4) (2011) 1239–1252.

[4] R. Bevilacqua, M. Romano, O.A. Yakimenko, Online generation of quasi-optimal spacecraft rendezvous trajectories, Acta Astronaut. 64 (2009) 345–358. (2–3).

[5] R. Bevilacqua, J. Hall, J. Horning, M. Romano, Ad-hoc wireless networking and shared computation based upon linux for autonomous multi-robot systems, AIAA J. Aerosp. Comput. Inf. Commun. 6 (5) (2009) 328–353.

[6] J. Hall, M. Romano, Robotic Testbed for the Experimentation of Guidance and Control of Spacecraft During Proximity Maneuvers. Mechatronic Systems, In-Tech, Vienna, Austria, 2010.

[7] R. Bevilacqua, M. Romano, F. Curti, A.P. Caprari, V. Pellegrini, Guidance navigation and control for autonomous multiple spacecraft assembly: analysis and experimentation, Int. J. Aerosp. Eng. (2011). (Article ID 308245).

[8] S.B. McCamish, M. Romano, X. Yun, Autonomous distributed control of simultaneous multiple spacecraft proximity maneuvers, IEEE Autom. Sci. Eng. 7 (3) (2010) 630–644.

[9] R. Bevilacqua, T. Lehmann, M. Romano, Development and experimentation of LQR/APF guidance and control for autonomous proximity maneuvers of multiple spacecraft, Acta Astronaut. 68 (6–7) (2011) 1260–1275.

[10] S.B. McCamish, M. Romano, S. Nolet, C.M. Edwards, D.W. Miller, Flight testing of multiple spacecraft control on SPHERES during close proximity operations, AIAA J. Spacecraft Rockets 46 (6) (2009) 1202–1213.

[11] M. Romano, D.A. Friedman, T.J. Shay, Laboratory experimentation of autonomous spacecraft approach and docking to a collaborative target, J. Spacecraft Rockets 44 (1) (2007) 164–173.

[12] O.A. Yakimenko, Direct method for rapid prototyping of near-optimal aircraft trajectories, J. Guid. Control Dynam. 23 (5) (2000) 865–875.

[13] G.A. Boyarko, M. Romano, O.A. Yakimenko, Time-optimal reorientation of a spacecraft using an inverse dynamics optimization method, J. Guid. Control Dynam. 34 (4) (2011) 1197–1208.

[14] A. Miele, H.Y. Huang, J.C. Heideman, Sequential gradient-restoration algorithm for the minimization of constrained functions—ordinary and conjugate gradient versions, J. Optim. Theory Appl. 4 (4) (1969) 213–643.

[15] C. Lugini, M. Romano, A ballistic-pendulum test stand to characterize small cold-gas thruster nozzles, Acta Astronaut. 64 (5–6) (2009) 615–625.